



**GALGOTIAS**  
UNIVERSITY

**School of Computing  
Science and Engineering**

Program: BSC (Hons) CS

Course Code: BSCS3560

Course Name: Linux Administration

Lecture : 12

## **UNIT II**

# **MONITORING AND MANAGING LINUX PROCESS AND LOGS**

- Linux process - Controlling Jobs - Background Process and Foreground Process - Monitoring Process Activity - Killing Processes - Reviewing syslog files.

# Process

- An instance of a program is called a Process. In simple terms, any command that you give to your Linux machine starts a new process.
- A **process** refers to a program in execution; it's a running instance of a program. It is made up of the program instruction, data read from files, other programs or input from a system user.
- **Types of Processes**
- Foreground Processes
- Background Processes

# Running a Foreground Process

- To start a foreground process, you can either run it from the dashboard, or you can run it from the terminal.
- When using the Terminal, you will have to wait, until the foreground process runs.



OR

```
home@VirtualBox:~$ banshee
```

# Running a Background Process

- If you start a foreground program/process from the terminal, then you cannot work on the terminal, till the program is up and running.
- Particular, data-intensive tasks take lots of processing power and may even take hours to complete. You do not want your terminal to be held up for such a long time.
- To avoid such a situation, you can run the program and send it to the background so that terminal remains available to you.

## LINUX Process

Start the program and press ctrl+z

```
guru99@VirtualBox:~$ banshee  
[Info 16:08:36.688] Running Banshee 2.2.1: [Ubuntu 11.  
11-12-19 14:51:26 UTC]  
^Z  
[1]+  Stopped                  banshee
```

Type 'bg' to send the process to the background

```
guru99@VirtualBox:~$ bg
```

## Fg

- You can use the command "fg" to continue a program which was stopped and bring it to the foreground.
- The simple syntax for this utility is:
- Fg jobname

### Example:

- Launch 'banshee' music player
- Stop it with the 'ctrl +z' command
- Continue it with the 'fg' utility.

# fg

```
home@VirtualBox:~$ banshee
^Z
[1]+  Stopped                  banshee
home@VirtualBox:~$ fg banshee
banshee
[Info 00:36:19.400] Running Banshee 2.2.0: [Ubuntu oneiric
(linux-gnu, i686) @ 2011-09-23 04:51:00 UTC]
```



## Top Command in Linux

- This utility tells the user about all the running processes on the Linux machine.

```
home@VirtualBox:~$ top
top - 23:57:43 up 2:54, 1 user, load average: 0.00, 0.01, 0.05
Tasks: 189 total, 2 running, 187 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.7%us, 3.0%sy, 0.0%ni, 96.3%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1026080k total, 924508k used, 101572k free, 37000k buffers
Swap: 1046524k total, 21472k used, 1025052k free, 367996k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1525	home	20	0	1775m	100m	28m	S	1.7	10.0	5:05.34	Photoshop.exe
961	root	20	0	75972	51m	7952	R	1.0	5.1	2:23.42	Xorg
1507	home	20	0	7644	4652	696	S	1.0	0.5	2:42.66	wineserver
1564	home	20	0	75144	29m	9840	S	0.3	3.0	0:25.96	ubuntuone-syncd
2999	home	20	0	127m	13m	10m	S	0.3	1.4	0:01.36	gnome-terminal
3077	home	20	0	2820	1188	864	R	0.3	0.1	0:00.76	top
1	root	20	0	3200	1704	1260	S	0.0	0.2	0:00.98	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.95	ksoftirqd/0

The field description are as follows:

Field	Description	Example 1	Example 2
PID	The process ID of each task	1525	961
User	The username of task owner	Home	Root
PR	Priority Can be 20(highest) or -20(lowest)	20	20
NI	The nice value of a task	0	0
VIRT	Virtual memory used (kb)	1775	75972
RES	Physical memory used (kb)	100	51
SHR	Shared memory used (kb)	28	7952
S	<p>Status</p> <p>There are five types:</p> <ul style="list-style-type: none"> <li>'D' = uninterruptible sleep</li> <li>'R' = running</li> <li>'S' = sleeping</li> <li>'T' = traced or stopped</li> <li>'Z' = zombie</li> </ul>	S	R
%CPU	% of CPU time	1.7	1.0
%MEM	Physical memory used	10	5.1
TIME+	Total CPU time	5:05.34	2:23.42
Command	Command name	Photoshop.exe	Xorg

## PS Command

- This command stands for 'Process Status'. It is similar to the "Task Manager" that pop-ups in a Windows Machine when we use Cntrl+Alt+Del. This command is similar to 'top' command but the information displayed is different.
- To check all the processes running under a user, use the command -

```
$ ps
```

PID	TTY	TIME	CMD
18	pts/1	00:00:00	sh
22	pts/1	00:00:00	ps

## Ps

- For more information -f (full) can be used along with ps

```
$ ps -f
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
52471	19	1	0	07:20	pts/1	00:00:00	f sh
52471	25	19	0	08:04	pts/1	00:00:00	ps -f

## PS

- For a single process information, `ps` along with process id is used

```
$ ps 19
PID  TTY      TIME    CMD
19   pts/1    00:00:00  sh
```

For a running program (named process) **Pidof** finds the process id's (pids)

## Fields described by ps are described as:

**UID:** User ID that this process belongs to (the person running it)

**PID:** Process ID

**PPID:** Parent process ID (the ID of the process that started it)

**C:** CPU utilization of process

**STIME:** Process start time

**TTY:** Terminal type associated with the process

**TIME:** CPU time taken by the process

**CMD:** The command that started this process

- There are other options which can be used along with ps command :

**-a:** Shows information about all users

**-x:** Shows information about processes without terminals

**-u:** Shows additional information like -f option

**-e:** Displays extended information



Thank You